MVVM框架的组成：数据绑定/指令系统/组件式编程/路由和导航/状态保持/第三方组件库

2. Angular框架

Angular v1.x官网：<https://angularjs.org/>

Angular v2.x～v8.x官网：<https://angular.io/>

Angular中文镜像网站：<https://angular.cn/>

是google2009年开发的MVVM框架

Vue.js适合中小型项目, Angular适合中大型项目

Web worker/web socket是html5中最有价值的，表单是很low的

1. 开发方式

Vue.js开发方式：script引入vue.js，脚手架方式

Angular2.x开发方式：脚手架方式

1. 搭建angular开发环境

前提：要有node.js

查看当前默认的下载仓库地址 npm config get registry

修改NPM默认的下载仓库地址为国内镜像网站：

npm config set registry 新仓库地址

1. 下载安装脚手架

npm install -g @angular/cli

1. 运行脚手架创建空白项目

ng new myngapp01

项目可以选想装在哪里

1. 运行开发服务器

Cd myngapp01

Npm start

1. 客户端访问测试

https://127.0.0.1:port

1. Angular的组件不是直接放在最外层的而是各个模块中
2. Angular项目启动过程分析：
3. **Angular.json**: NG项目的配置

Index: ./src/index.html <app-root></app-root>

Main: ./src/main.ts

1. **Main.js** > AppComponent
2. **App.component.ts** > selector: ‘app-root’

* templateUrl: ‘app.component.html’

1. **app.component.html** > html片段
2. 一个组件由三个组成：templateurl，script，style
3. Angular核心概念之一：模块

Module：不同于Node.js或者ES6中的模块的模块，NG中的模块就是一个抽象的容器，用于对组件分组。

整个应用初始时有且只有一个主模块：AppModule

1. Angular核心概念之二：组件

组件：是一段可以反复使用的页面片段，如页头/轮播/手风琴等

组件（component）= 模版（template）+脚本（script）+样式（style）

提示：在NG中任何一个组件都需要声明/注册在一个模块中

1. 自定义组件步骤：
2. 创建组件class

@Component({

template: '<h2>我的组件C01</h2><hr>',

selector: '[myTitle]' /\*属性来用要加[]，当元素来用直接用没有[]，如果.xxx，当作class来用 \*/

})

export class MyC01Component {

}

1. 在某个模块中注册组件class

如果没有找到注册的文件app.module.ts的话，创建项目的时候要加一些指令

ng new my-app --no-standalone --routing --ssr=false

1. 使用组件

//app.component.html

<myc01></myc01>

练习：创建一个自定义组件myc02，创建class，注册，使用

练习：创建一个文件夹，myc03，其中在自定义组件myc03.component.ts

1. **Angular提供的创建组件的简化工具**

ng g[enerate] component 组件名

npx ng g[enerate] component 组件名

**\*\* .spec.ts是做单元测试的，不是一定要用**

1. Node.js官方安装的工具：

npm：第三方模块的维护工具

npx：第三方可执行文件的执行工具，Node Package Executor npx 可用于执行当前项目中node\_modules./bin目录下的可执行

1. Angular核心概念之三：数据绑定
2. Html绑定：{{NG表达式}}

测试：NG表达式中可以执行哪些代码？

算术运算？

比较运算：

逻辑运算：

三目运算：

调用函数：

创建对象：不可以new

<!-- <div>当前时间：{{new Date()}}</div> -->

Json序列化：不可以，因为NG表达式中JSON是undefined

<!-- <div>JSON字符串：{{JSON.stringify({})}}</div> -->

1. 属性绑定：Vue.js：v-bind

形式1：直接在属性上用{{}}绑定

<p title="{{uname}}">

形式2:（推荐）直接使用中括号[]

<p [title]="uname">

1. 事件绑定：Vue.js：v-on或者简写为@
2. 指令绑定
3. 双向数据绑定
4. 课后任务：

（1） 比较vue.js和angular中的表达式有何不同

（2） 自学手册中的“事件绑定”和“指令绑定”，实现购买计数器n

Day2

1. Angular中的数据绑定
2. HTML绑定：{{}}
3. 属性绑定：[]

注意：如果赋值是常量要加单引号

1. 事件绑定

(click)=”add()”

1. 指令绑定
   1. **循环绑定：\*ngFor**

<Any \*ngFor=”let 临时变量 of 数据”>

<Any \*ngFor=”let 临时变量 of 数据; let i=index;”>

<Any \*ngFor=”let 临时变量 of 数据; index as i””>

* 1. **选择绑定： \*ngif**

<Any \*ngif=”布尔表达式”>

说明：如果布尔表达式为false，则当前元素从DOM树上删除

<Any \*ngif=”布尔表达式； else ELSE块的编号”></Any>

<ng-template #ELSE块的编号>

<Any></Any>

</ng-template>

* 1. **样式绑定**

ngStyle/ngClass

说明：ngStyle绑定的值必须是一个对象！对象属性就是CSS样式名

ngClass

说明：ngClass绑定的值必须是一个对象！对象属性就是CSSclass名，属性值是true/false的话该class就出现，否则该class不出现

* 1. 了解：特殊的选择绑定：

<Any [ngSwitch]=’表达式’>

<Any \*ngSwitchCase=”值1”></Any>

<Any \*ngSwitchCase=”值2”></Any>

<Any \*ngSwitchDefault=”值3”></Any>

1. 双向数据绑定：[(ngModel)]-重点

方向一：Model => View, 模型变则视图变，用[]绑定

方向二：View => Model，视图（表单元素）变则模型变，用()绑定

<input/select/textarea [(ngModel)]=”uname”>

注意：如果想要监听模型数据改变，可以绑定ngModel指令不在CommonModule模块中，而在FormsModule中，使用之前必须在主模块中导入该模块；ngModel指令不在CommonModule模块中，而在FormsModule中，使用之前必须在主模块中导入该模块：

imports: [

BrowserModule,//if you use Angular to create a web application, you should have this module, including ngFor/ngIf, not including ngModel

AppRoutingModule,

FormsModule

],

1. 练习：注册新用户时，密码字段随着输入的进行，后面span中显示提示文字，可能为：

密码长度应该在6～12位之间

密码太短了，

密码长度合适

密码太长了

Angular中的指令分为3类，什么时候用\*什么时候用[]：

1. 组件指令：NG中Component继承自directive

Eg. ng g directive 指令名称

（2）结构型指令：会影响DOM树结构，必须使用\*开头，如\*ngFor, \*ngIf

（3）属性型指令：不会影响DOM树结构，只是影响元素外观或者行为，必须用[]括起来，如[ngClass],[ngStyle]

1. 练习：创建变量保存银行储户基本 userLevel = “normal”/”vip”/”blackgold”，根据用户级别不同，看到不同的内容，如果用户基本不在上面三个之一，呈现“欢迎您：游客”

Angular-5.1-复习

1. 练习：创建员工列表数据： empList=[{eid:101,ename:’亮亮’,sex1,zzmm:10, salary:5000}],把上述数据绑定在table中，最后添加一列<button>删除</button>，点击后删除该员工

Angular-5.5过滤器-01

Filter:过滤器，用于在View中呈现数据时显示为另一种格式；过滤器的本质是一个函数，接收原始数据转换为新的格式进行输出function（oldVal）{return newVal}

使用过滤器：{{e.salary | 过滤器名}}

Angular2.x中，过滤器更名为“管道（pipe）”

自定义管道的步骤：

1. 创建管道class，实现转换功能

@Pipe({

name: "sex" //过滤器/管道的名字

})

export class SexPipe {

//管道中执行过滤任务的是一个固定的函数

transform(val:number){

if(val==1) {

return '男'

}else if(val==0) {

return '女'

}else{

return '未知'

}

}

}

1. 在模块中注册管道

//app.module.ts

Declarations:[SexPipe]

1. 在模版视图中使用管道

{{e.empSex|sex}}

也可以用到属性里

<p title=”empSex|sex”>

说明：管道的transform方法除了val还可以接受其他参数，调用管道时冒号为这些参数赋值，例如：{{e.empSex|sex:’en’}}

创建管道的工具：ng g pipe zzmm

1. 创建一个政治面貌转换过滤器/管道ZzmmPipe，可以接收一个可选参数，format，转换规则如下：

如果format 值为‘short’，则转换规则为10-党员，20-团员，30-群众

如果format值为‘long’，则转换规则为10-中国共产党员，20-中国共青团员，30-中国人民群众

Vue.js中没有预定义管道，Angular有

1. Lowercase：转换成小写

<td>{{e.ename|lowercase}}</td>

1. Uppercase：转换成大写

<td>{{e.ename|uppercase}}</td>

1. Titlecase：转换成首字母大写

<td>{{e.ename|titlecase}}</td>

Note: 两个word要分开才能识别两个分别首字母大写

1. Slice：只显示字符串的一部分

<td>{{e.ename|slice:0:3}}</td>

1. Json：把js对象序列化成json字符串，在调试的时候很有用
2. Number：把数字转换为具有指定整数位和小数位的字符串

{{num|number}}

{{num|number:’4.1-4’}}

1. Currency：把数字转换成货币格式字符串：货币符号+三位一逗号+二位小数位

{{num|currency}}

{{num|currency:’¥’}}

1. Date：把数字转换成日期字符串

{{num|date:’yyyy-MM-dd HH:mm:ss’}}

1. 创建对象的两种方式

方式一：手工创建式 – 自己创建：let c2 = new Car()

方式二：依赖注入式 – 无需自己new，只需要声明依赖；服务提供者就会创建被依赖的对象，注入给服务需要者

1. Angular核心概念之六 – 服务和依赖注入 – 抽象但重点！

Service：服务，跟用户有关的放在组件（跟用户交互的一种对象）里，其中的内容都应该与用户操作有关系的，而跟用户操作无关的都应该剥离出去，放在“服务对象”中，为组件服务；例如：日志记录，计时统计，数据服务器的访问。

但vue.js不一样，他没有服务的概念，都是放在组件里，所以vue.js是中小型框架，angular是大型框架

依赖注入

Dependency injection

组件

服务提供者

创建

Instance

声明依赖

Dependency

服务